

移动平台应用开发

—— 手机通讯录软件

实 验 报 告

学 院 计控学院
年 级 11 级
班 级 计算机科学与技术
学 号 1110568
姓 名 郭艺璇

2014 年 5 月 28 日

目录

一、实验目的	1
二、实验内容	1
2.1 实验要求:	1
2.2 实验完成情况.....	1
三、实验步骤及实验结果	2
3.1 项目成果.....	2
3.2 项目 UI 设计.....	7
3.2 项目类文件设计.....	9
四、实验遇到的问题及其解决方法	17
4.1 页面间传值问题.....	17
五、实验结论	18

一、实验目的

- 1、本次实验的目的是让大家熟悉 Android 中对数据库进行操作的相关的接口、类等。SQLiteDatabase 这个是在 android 中数据库操作使用最频繁的一个类。通过它可以实现数据库的创建或打开、创建表、插入数据、删除数据、查询数据、修改数据等操作。
- 2、实现添加用户名，爱好小例程。

二、实验内容

2.1 实验要求：

利用学习的 Intent、Service 以及 SQLiteDatabase 创建并实现一个简易通讯录，要求：

1. 要有增、删、改，以及简单的搜索功能。
2. 通讯录需要有如下表单：姓名、手机、电话、邮箱、地址（或更多）。
3. 所有数据都存储在 SQLite 数据库中。
4. 在项目中要用到 Activity，Intent（至少要有有一个页面跳转），SQLiteDatabase 或 SQLiteOpenHelper。查看内容要以列表的形式体现，添加 5 个（或更多）联系人。

简洁大方的 UI，并且具备基本的错误判断（如对用户操作的简单提醒，对用户错误操作的警示等）。

2.2 实验完成情况

实验项目文件：6 个布局文件和 10 个 JAVA 文件。

- 1、联系人的基本信息包括：姓名，头像，备注，手机，座机，邮箱，地址，公司、博客 9 个信息。
- 2、实现了增、删、改、查四个基本功能。
- 3、所有的数据存储在 SQLite 数据库当中。
- 4、项目当中用到了 Activity 和 Intent 跳转，采用 SQLiteDatabase 和 SQLiteOpenHelper 来进行数据库的操作，查看的内容以列表的形式显示了头像、姓名和电话三个信息，点击后跳转到详细信息页面。
- 5、具有简洁大方的 UI，对用户输入具有一定的判断能力。电话和座机只允

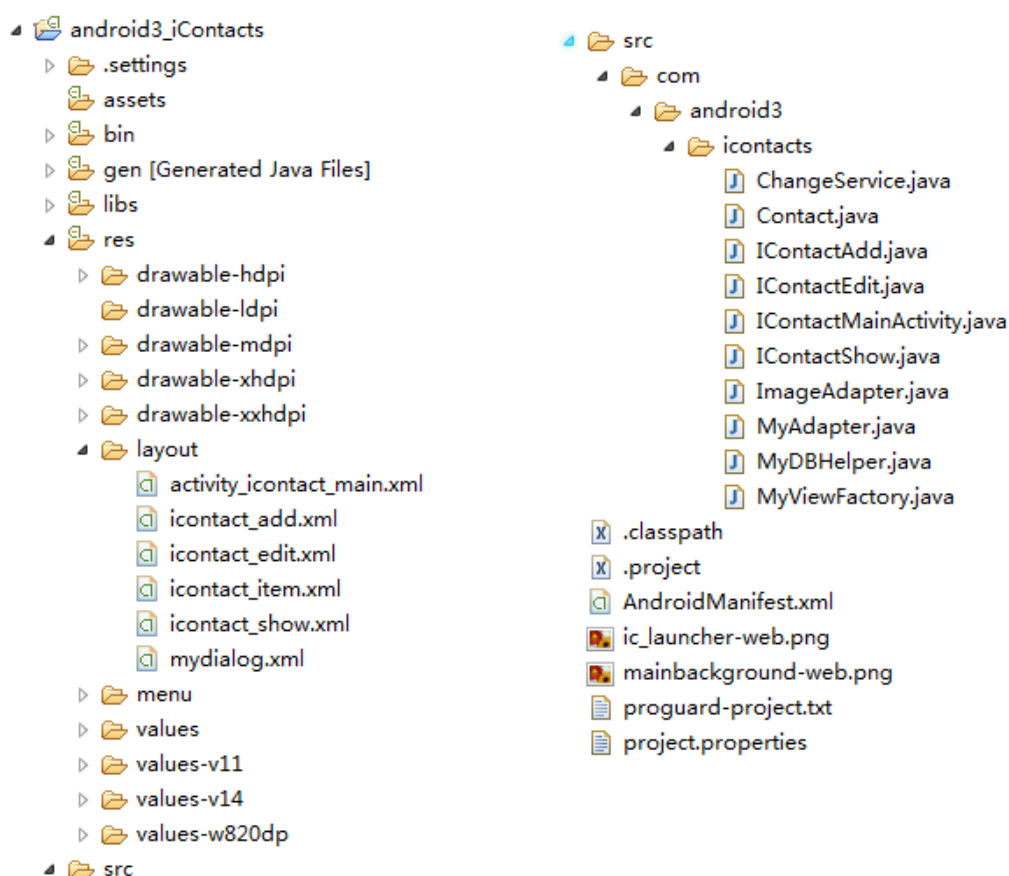
许输入数字（最多 11 位），座机和手机必须有一个输入为有效号码（中国标准为 11 位数字），否则提示用户重新输入信息。

6、拓展的功能包括：

6.1 头像功能：用户可以根据 gallery 列出的一系列头像选择联系人的头像，并且头像 id 保存在数据库中，在主界面的用户信息行和用户详细信息叶中会显示，头像也可以进行修改。

6.2 自动匹配邮箱功能：用户输入邮箱时，会产生下拉框显示自动匹配各种邮箱后缀，用户可直接点击正确的邮箱，不必输入后缀。

6.3 删除操作确认与提醒功能：用户删除数据时会弹出窗口确认删除，避免误删现象，删除成功后提示用户。



三、实验步骤及实验结果

3.1 项目成果

本次实验，实现了一个功能较为完全的通讯录，具有显示联系人列表的主界面，添加联系人界面，显示联系人详细信息界面和修改联

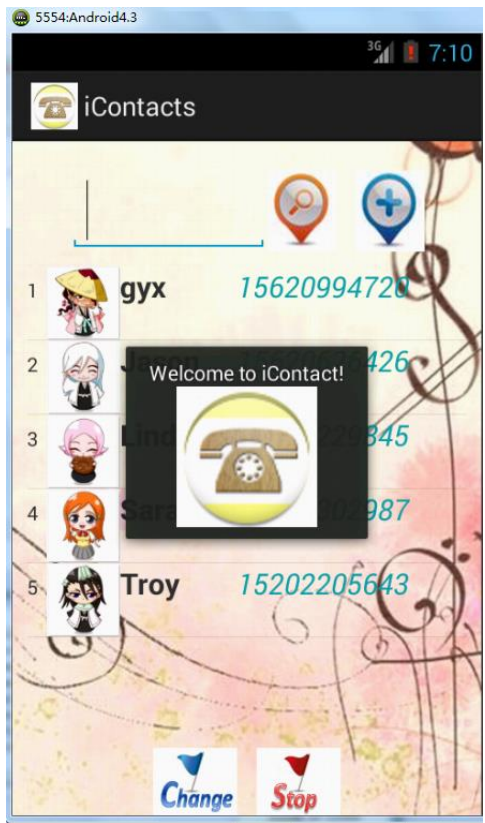


iContacts

系人信息界面。 项目的图标和名字 iContacts 如右图：
软件的整体操作流程如下图所示：



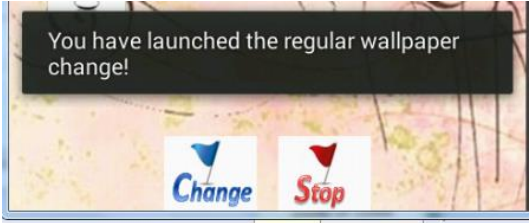
以下是具体操作流程和界面效果。点击软件图标，进入主界面，显示“Welcom to iContact!” 欢迎字样，通讯录中存储的联系人按行显示，包括联系人所在数据库中 id,联系人头像，名字和电话：



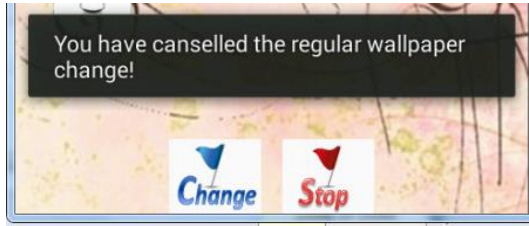
头部有搜索框和搜索按钮，添加联系人按钮。

底部的 Change 和 Stop 为开启定期修改壁纸和停止定期修改的按钮，为基于 service 的操作，点击后会有提示。

点击 Change 按钮：

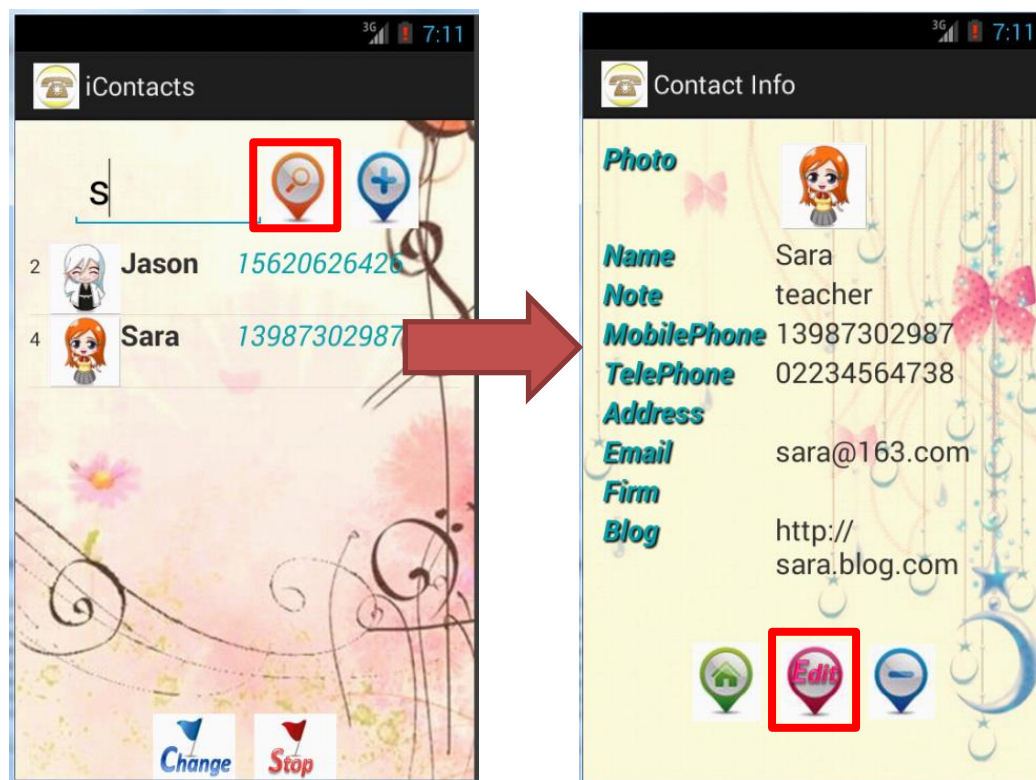


点击 Stop 按钮：



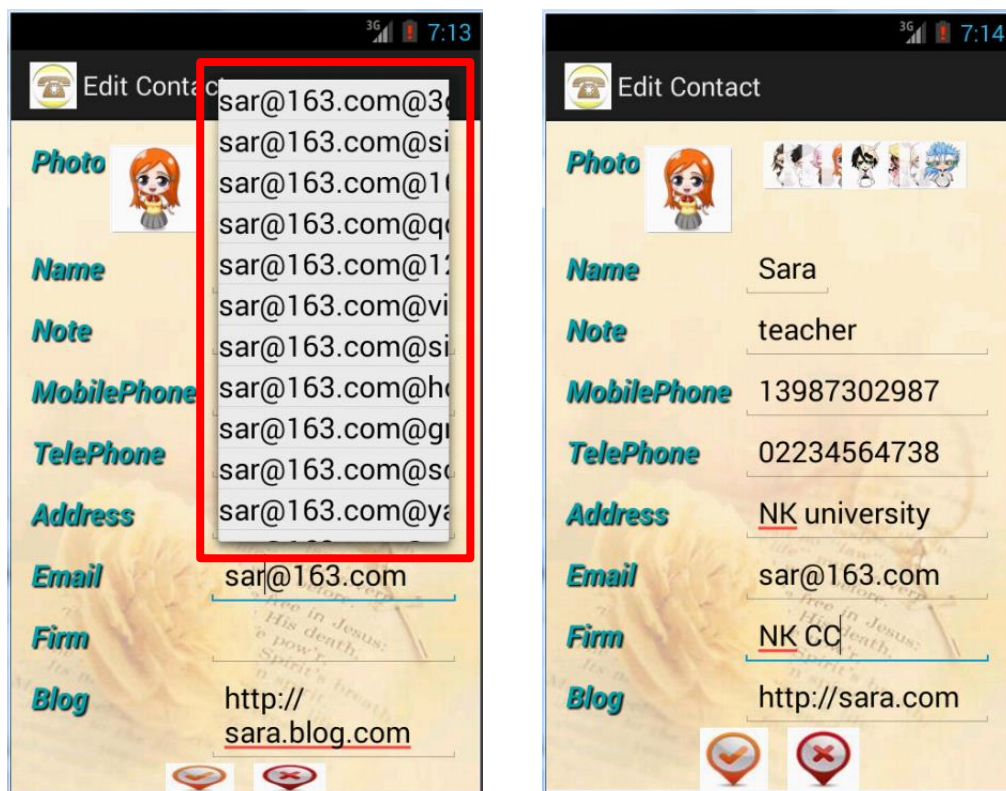
在搜索框中输入“s”字符，点击搜索按钮，则会进行按名字搜索，名字中带有该字符的会即时显示在界面上。

点击联系人，进入联系人详细信息显示界面，这里点击名字为“Sara”的联系人。显示出“Contact Info”界面，罗列出 Sara 的详细信息，有三个按钮，可以返回主界面，编辑该联系人，或者删除该联系人。



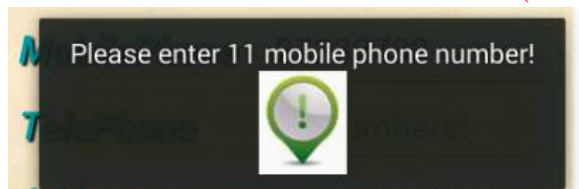
点击“Edit”按钮，显示编辑界面，可以对所有信息进行编辑：

[其中：Photo 可以通过点击一个 gallery 进行选择，在填写 email 时输入字符后会自动提示各种邮箱的后缀，方便用户直接进行选择。]



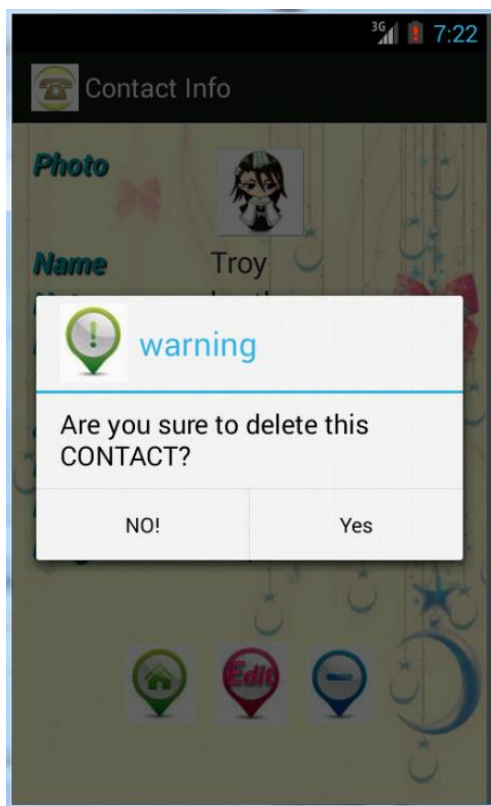
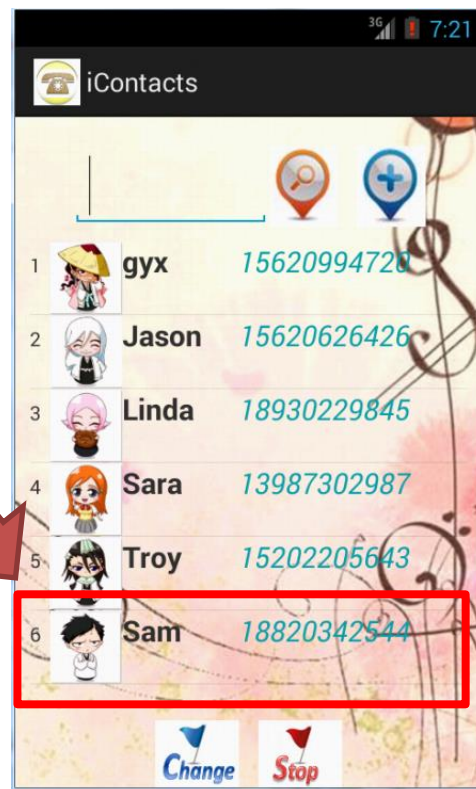
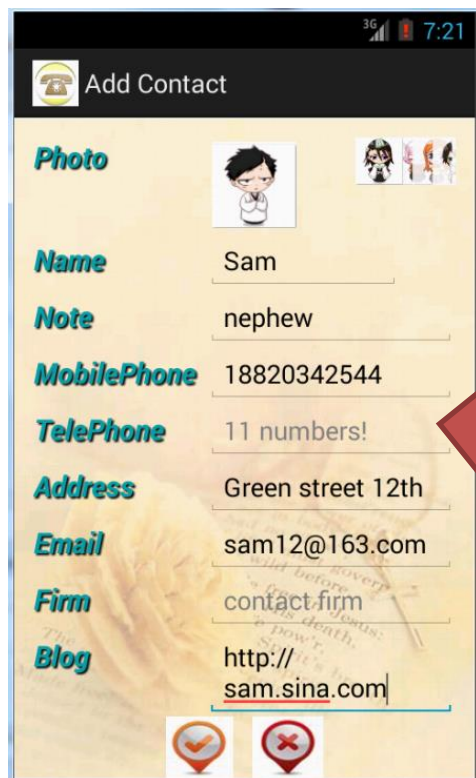
修改好之后，点击确认按钮，跳转到主界面，此时数据库中“Sara”的数据已经进行更新。

在主界面点击添加按钮，显示添加界面，我们输入信息进行添加。需要注意的是，此处对 Mobile phone 和 telephone 作了限制，最大输入字数为 11 个，且只能输入数字，在添加前进行判断，两个电话必须有一个是 11 位的有效电话号码(中国的手机和座机号码均为 11 位)，否则提示用户 “Please enter 11 mobile phone number!”

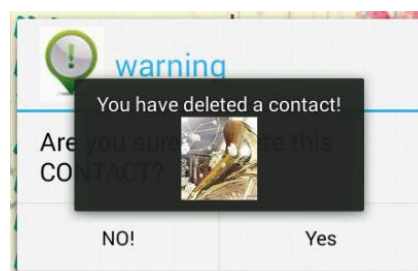


如果我们选择取消按钮，则不进行数据插入操作，返回主界面。

我们选择确认键，数据插入后返回主界面，可以看到新插入的数据显示在 listview 当中。

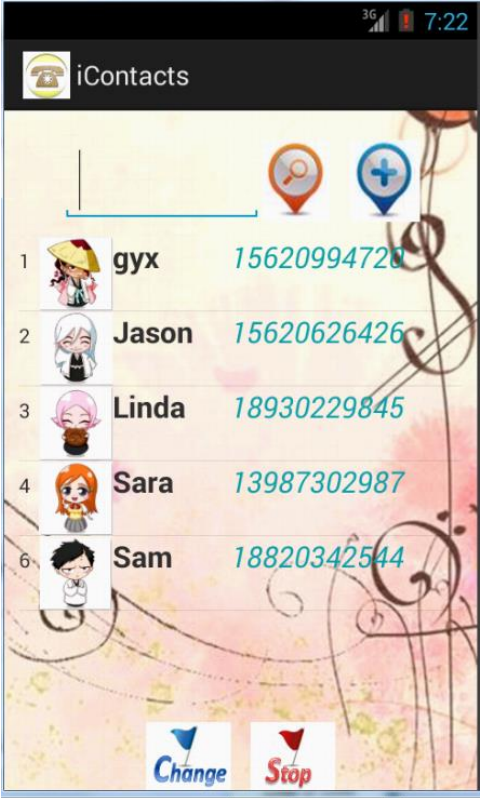


删除操作在数据显示界面进行,我们从主界面选择名字为“Troy”的联系人点击,进入联系人数据显示界面,点击删除按钮,弹出提示框,提示用户“Are you sure to delete this CONTACT?”若选择“NO!”则不删除用户,若选择“Yes”,从数据库删除该用户数据,提示框提示用户已经成功删除了一个联系人。



之后返回到主界面上，可以看到“Troy”联系人的数据已经删除了。

以上就是通讯录的整体效果。



3.2 项目 UI 设计

本项目采用了 4 个主要的界面，分别是主界面

activity_icontact_main.xml，添加联系人界面 icontact_add.xml，显示联系人界面 icontact_show.xml，和修改联系人界面 icontact_edit.xml。

其中，主界面最为复杂，包括 1 个搜索框 EditText，4 个 ImageButton（搜索功能、添加功能、开启壁纸更换和关闭壁纸更换功能）和 1 个 ListView（显示联系人基本信息）。ListView 的每行的内容在 icontact_item.xml 当中，如右所示：

```
activity_ic...  icontact_it...  IContactMain...
<TextView <!-- 数据库中id号 -->
    android:layout_weight="0"
    android:layout_width="20px"
    android:layout_height="wrap_content"
    android:textSize="20px"
    android:id="@+id/iid"
/>
<TextView <!-- 头像所对应的图片id. 不显示 -->
    android:id="@+id/iimgt"
    android:layout_width="0dp"
    android:layout_height="0dp"/>
<ImageView <!-- 头像 -->
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:textSize="20px"
    android:id="@+id/iimg"/>
<TextView <!-- 名字 -->
    android:layout_width="80dp"
    android:textStyle="bold"
    android:layout_height="wrap_content"
    android:textSize="30px"
    android:id="@+id/iname" />
<!-- 电话 -->
<TextView
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:textStyle="italic"
    android:textColor="#ff0aa0aa"
    android:textSize="30px"
    android:id="@+id/imobph"/>
```

icontact_show.xml, icontact_add.xml, icontact_edit.xml 三个界面均为联系人的 9 项详细信息的列表, show 界面仅有 TextView 展示信息, add 和 edit 界面则有 EditText 可以输入信息, 每个界面有各自的 ImageButton 支持不同的操作。

```
IContactMain...  icontact_sh...  IContactEdi...  IContactShow...
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:gravity="center">
    <ImageButton
        android:id="@+id/backbn"
        android:scaleType="fitXY"
        android:background="@drawable/icon_back"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"></ImageButton>
    <ImageButton
        android:layout_marginLeft="20px"
        android:id="@+id/editbn"
        android:scaleType="fitXY"
        android:background="@drawable/icon_edit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />
    <ImageButton
        android:layout_marginLeft="20px"
        android:id="@+id/delbn"
        android:scaleType="fitXY"
        android:background="@drawable/icon_del"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />
</LinearLayout>
```

show 界面有返回主界面, 编辑和删除三个 ImageButton: add 界面和 edit 界面均有确定和取消两个按钮。

```
icontact_edi...  activity_ic...  icontact_it...  IContactMain...
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:gravity="center">
    <ImageButton
        android:id="@+id/okbn2"
        android:scaleType="fitXY"
        android:layout_height="wrap_content"
        android:background="@drawable/icon_ok"
        android:layout_width="wrap_content"></ImageButton>
    <ImageButton
        android:layout_marginLeft="20px"
        android:id="@+id/clbn2"
        android:scaleType="fitXY"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/icon_cl"
    />
</LinearLayout>
```

3.2 项目类文件设计

3.2.1 新建 MyDBHelper.java 文件，用于对接口化数据库操作，调用后自动新建“icontact.db”数据库和“icontactTbl”表，数据库的插入、删除、查询操作可以很方便地直接调用对应的成员函数：

```
public class MyDBHelper extends SQLiteOpenHelper{
    //数据库名, 表名
    private static final String DB_NAME="icontact.db";
    private static final String TBL_NAME="icontactTbl";
    //创建表的语句
    private static final String
        CREATE_TBL="CREATE TABLE "+ "icontactTbl (_id INTEGER DEFAULT '1' NOT NULL"+" PRIMARY KEY AUTOINCREMENT,"
        + "img INTEGER, name TEXT, note TEXT,mobph TEXT, telph TEXT, addr TEXT, email TEXT, firm TEXT, blog TEXT)";

    //SQLiteDatabase实例
    private SQLiteDatabase db;

    //构造方法
    public MyDBHelper(Context context){
        super(context, DB_NAME, null, 2);
        System.out.printf("创建数据库");
    }

    //创建表
    public void onCreate (SQLiteDatabase db){
        this.db=db;
        db.execSQL(CREATE_TBL);
        System.out.printf("创建表");
    }

    //插入方法
    public void insert(ContentValues values){
        //获得SQLiteDatabase实例
        SQLiteDatabase db=getWritableDatabase();
        //插入
        db.insert(TBL_NAME, null, values);
        db.close();
        System.out.printf("数据库操作");
    }

    //查询方法
    public Cursor query(){
        System.out.printf("数据库查询方法");
        //获得SQLiteDatabase实例
        SQLiteDatabase db=getWritableDatabase();
        //查询获得Cursor
        Cursor c=db.query(TBL_NAME, null, null, null, null, null, null);
        return c;
    }

    //删除方法
    public void del(int id){
        System.out.printf("数据库删除方法");
        if(db==null){
            SQLiteDatabase db=getWritableDatabase();
            db.delete(TBL_NAME, "_id=?", new String[]{String.valueOf(id)});
        }
    }

    //关闭数据库
    public void close(){
        System.out.printf("数据库删除方法");
        if(db!=null){
            db.close();
        }
    }

    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
    {
        db.execSQL("DROP TABLE IF EXISTS " + TBL_NAME);
        onCreate(db);
    }
}
```

3.2.2 修改 IContactMainActivity.java 文件，这是程序启动后第一个调用的 activity，该程序主要有如下几个功能：

查询并显示数据库中联系人：

```
IContactMain...  MyDBHelper.java  icontact_ad...  icontact_edi...  activity_ic...  ico
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_icontact_main);

    Toast toast = Toast.makeText(getApplicationContext(), "Welcome to iContact!", 100);
    toast.setGravity(Gravity.CENTER, 0, 0);
    LinearLayout toastView = (LinearLayout) toast.getView();
    ImageView iv = new ImageView(getApplicationContext());
    iv.setImageResource(R.drawable.ic_launcher);
    toastView.addView(iv);
    toast.show();

    mylistview = (ListView) findViewById(R.id.listview);
    MySQLite();
}

private void MySQLite()
{
    MyDBHelper helper = new MyDBHelper(this);

    SQLiteDatabase db = helper.getReadableDatabase();
    //查询数据库里的数据
    //helper.onUpgrade(db, 1, 1);
    Cursor c = db.query("icontactTbl", new String[]{"_id", "img", "name", "mobph"}, null, null, null, null, null);
    list = new ArrayList<Map<String, Object>>();
    while(c.moveToNext())
    {
        Contact modle = new Contact();
        map = new HashMap<String, Object>();
        map.put("id", modle.setid(c.getInt(c.getColumnIndex("_id"))));
        map.put("img2", modle.setimg(c.getInt(c.getColumnIndex("img"))));
        map.put("img", modle.setimg(c.getInt(c.getColumnIndex("img"))));
        //map.put("img2", "img");
        map.put("name", modle.setName(c.getString(c.getColumnIndex("name"))));
        map.put("mobph", modle.setmobph(c.getString(c.getColumnIndex("mobph"))));
        list.add(map);
    }
    //创建SimpleAdapter:适配器将数据绑定到item显示控件上
    adapter = new SimpleAdapter(IContactMainActivity.this, list, R.layout.icontact_item,
        new String[]{"id", "img2", "img", "name", "mobph"}, new int[]{R.id.iid, R.id.iimgt, R.id.iimg, R.id.iname, R.id.imobph});

    mylistview.setAdapter(adapter);
    db.close();
}
```

搜索和即时显示姓名匹配联系人，通过查询语句得到匹配字符串的联系人数据并同步显示出来：

```

ImageButton srch_bn=(ImageButton)findViewById(R.id.srchbn);
srch_bn.setOnClickListener(new OnClickListener(){
    @SuppressWarnings("deprecation")
    public void onClick(View v){
        final EditText searchText = (EditText)findViewById(R.id.srchtxt);
        SimpleCursorAdapter adapter1 = null;
        content=searchText.getText().toString().trim();

        dbhp=new MyDBHelper(IContactMainActivity.this);
        sqlDB=dbhp.getReadableDatabase();
        cursor=sqlDB.rawQuery("select * from icontactTbl where name like '%" + content + "%'",null);
        list=new ArrayList<Map<String,Object>>();
        while(cursor.moveToNext())
        {
            Contact modle=new Contact();
            map=new HashMap<String,Object>();
            map.put("id", modle.setid(cursor.getInt(cursor.getColumnIndex("_id"))));
            map.put("img2", modle.setimg(cursor.getInt(cursor.getColumnIndex("img"))));
            map.put("img", modle.setimg(cursor.getInt(cursor.getColumnIndex("img"))));
            //map.put("img2", "img");
            map.put("name", modle.setName(cursor.getString(cursor.getColumnIndex("name"))));
            map.put("mobph", modle.setmobph(cursor.getString(cursor.getColumnIndex("mobph"))));
            list.add(map);
        }
        adapter=new SimpleAdapter(IContactMainActivity.this,list,R.layout.icontact_item,
            new String[]{"id","img2","img","name","mobph"},new int[]{R.id.iid,R.id.imgt,R.id.img,R.id.iname,R.id.imobp

mylistview.setAdapter(adapter);
sqlDB.close();
}
});

```

点击联系人基本信息栏，跳转到该联系人的详细信息界面，此处需要进行页面间传值，联系人在数据库中的 id 值和联系人头像的 id 值传递到信息显示页面当中，以便通过两个 id 值定位联系人信息和头像，进行正确显示：

```

mylistview.setOnItemClickListener(new OnItemClickListener() {

    public void onItemClick(AdapterView<?> parent, View view, int position,
        long arg3)
    {
        ListView listview=(ListView)parent;
        HashMap<String, Object> data = (HashMap<String, Object>) listview.getItemAtPosition(position);
        Intent intent = new Intent();
        TextView tvid = (TextView) view.findViewById(R.id.iid);
        TextView iv=(TextView)view.findViewById(R.id.imgt);
        intent.putExtra("id", tvid.getText().toString());
        intent.putExtra("img",Integer.parseInt(iv.getText().toString()));

        //Toast.makeText(IContactMainActivity.this, "onCreate()--imgID="+iv.getText().toString(), Toas

        intent.setClass(IContactMainActivity.this, IContactShow.class);
        startActivity(intent);
    }
});

```

点击添加联系人按钮，跳转到添加界面：

```

ImageButton add_bn=(ImageButton)findViewById(R.id.addbn);
add_bn.setOnClickListener(new OnClickListener(){
    public void onClick(View v){
        Intent intent_re=new Intent(IContactMainActivity.this,IContactAdd.class);
        startActivity(intent_re);
    }
});

```

更换壁纸的开始和停止功能：


```

change = (ImageButton) findViewById(R.id.changebn);
stop = (ImageButton) findViewById(R.id.stopbn);
aManager = (AlarmManager) getSystemService(
    Service.ALARM_SERVICE);
// 指定启动ChangeService组件
Intent intent = new Intent(IContactMainActivity.this,
    ChangeService.class);
// 创建PendingIntent对象
final PendingIntent pi = PendingIntent.getService(
    IContactMainActivity.this, 0, intent, 0);
change.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View arg0)
    {
        // 设置每隔5秒执行pi代表的组件一次
        aManager.setRepeating(AlarmManager.RTC_WAKEUP
            , 0, 5000, pi);
        change.setEnabled(false);
        stop.setEnabled(true);
        Toast.makeText(IContactMainActivity.this
            , "You have launched the regular wallpaper change!",
            Toast.LENGTH_SHORT).show();
    }
});

stop.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View arg0)
    {
        change.setEnabled(true);
        stop.setEnabled(false);
        // 取消对pi的调度
        aManager.cancel(pi);
        Toast.makeText(IContactMainActivity.this
            , "You have cancelled the regular wallpaper change!",
            Toast.LENGTH_SHORT).show();
    }
});

```

3.2.3 添加联系人类 IContactAdd.java

选择头像功能，通过 gallery 进行选择头像，点击一个头像后，将图片的 id 传给 imgID，利用 imgID 的值对 ImageView 进行赋值，显示图片，保存在数据库中的也使图片的 id:

```

gallery=(Gallery)findViewById(R.id.gallery_add);
imageview=(ImageView)findViewById(R.id.iv_img);
imageadapter=new ImageAdapter(this);
gallery.setAdapter(imageadapter);
gallery.setOnItemClickListener(new OnItemClickListener(){

    @Override
    public void onItemClick(AdapterView<?> parent, View v, int position,
        long id) {
        // TODO Auto-generated method stub
        imgID=(int)imageadapter.getItemId(position);
        imageview.setImageResource(imgID);
        imageview.setId(imgID);
    }
});

```

邮箱匹配功能，用户输入邮箱时，侦听 EditText 的输入信息，一旦信息改变则匹配邮箱后缀，方便用户输入：

```
String[] arrayString = new String[] { "@3g.sina.cn", "@sina.com",
    "@163.com", "@qq.com", "@126.com", "@vip.sina.com", "@sina.cn",
    "@hotmail.com", "@gmail.com", "@sohu.com", "@yahoo.com", "@tom.com" };
AutoCompleteTextView autoview;
autoview = (AutoCompleteTextView)findViewById(R.id.edemail);
final MyAdatper adapter=new MyAdatper(this);
autoview.setAdapter(adapter);
autoview.addTextChangedListener(new TextWatcher() {
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        // TODO Auto-generated method stub
    }
    public void beforeTextChanged(CharSequence s, int start, int count,
        int after) {
        // TODO Auto-generated method stub
    }
    public void afterTextChanged(Editable s) {
        // TODO Auto-generated method stub
        String input = s.toString();
        adapter.mList.clear();
        if (input.length() > 0) {
            for (int i = 0; i < arrayString.length; ++i) {
                adapter.mList.add(input+arrayString[i]);
            }
        }
        adapter.notifyDataSetChanged();
        autoview.showDropDown();
    }
});
autoview.setThreshold(1);
```

信息输入完毕，检查号码是否输入符合要求（11 位数字），检查无误则插入数据，跳转到主界面：

```
if(mobph.length()==11||telph.length()==11){//判断电话号码是否
    ContentValues values=new ContentValues();
    values.put("img", imgid);
    values.put("name", sname);
    values.put("note", snote);
    values.put("mobph", smobph);
    values.put("telph", stelph);
    values.put("addr", saddr);
    values.put("email", semail);
    values.put("firm", sfirm);
    values.put("blog", sblog);
    MyDBHelper helper=new MyDBHelper(IContactAdd.this);
    SQLiteDatabase db=helper.getWritableDatabase();
    db.insert("icontactTbl", null, values);
    db.close();
```

```
    Intent intent=new Intent();
    intent.setClass(IContactAdd.this, IContactMainActivity.class);
    startActivity(intent);
    finish();
}
```

```
else{
    Toast toast = Toast.makeText(getApplicationContext(), "Please enter 11 mobile phone number!", 30);
    toast.setGravity(Gravity.CENTER,0,0);
    LinearLayout toastView=(LinearLayout)toast.getView();
    ImageView iv=new ImageView(getApplicationContext());
    iv.setImageResource(R.drawable.icon_warn);
    toastView.addView(iv);
    toast.show();
}
}
});
```

```
okbn.setOnClickListener(new OnClickListener(){
    public void onClick(View v){
        int imgid=imgID;//imageview.getId();
        String sname=name.getText().toString();
        String snote=note.getText().toString();
        String smobph=mobph.getText().toString();
        String stelph=telph.getText().toString();
        String saddr=addr.getText().toString();
        String semail=email.getText().toString();
        String sfirm=firm.getText().toString();
        String sblog=blog.getText().toString();
```

3.2.4 显示联系人详细信息类 IContactShow.java

通过读取页面传递过来的联系人 id 和头像 id 信息，查询数据库，显示联系人详细信息和头像信息：

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.icontact_show);
    Bundle bundle=this getIntent().getExtras();
    passid=bundle.getString("id");
    passing=bundle.getInt("img");
    name=(TextView)findViewById(R.id.shname);
    note=(TextView)findViewById(R.id.shnote);
    mobph=(TextView)findViewById(R.id.shmobph);
    telph=(TextView)findViewById(R.id.shtelph);
    addr=(TextView)findViewById(R.id.shaddr);
    email=(TextView)findViewById(R.id.shemail);
    firm=(TextView)findViewById(R.id.shfirm);
    blog=(TextView)findViewById(R.id.shblog);
    backbn=(ImageButton)findViewById(R.id.backbn);
    editbn=(ImageButton)findViewById(R.id.editbn);
    delbn=(ImageButton)findViewById(R.id.delbn);

    imageview=(ImageView)findViewById(R.id.img_show);
    imageadapter=new ImageAdapter(this);
    imageview.setImageResource(passing);
    MyDBHelper helper=new MyDBHelper(IContactShow.this);
    SQLiteDatabase db=helper.getReadableDatabase();
    Cursor cursor=db.query("icontactTbl",
        new String[] {"name","note","mobph","telph","addr","email","firm","blog"},
        "_id=?", new String [] {passid}, null, null, null);
    while(cursor.moveToNext()){
        Contact modle=new Contact();
        name.setText(modle.setName(cursor.getString(cursor.getColumnIndex("name"))));
        note.setText(modle.setName(cursor.getString(cursor.getColumnIndex("note"))));
        mobph.setText(modle.setName(cursor.getString(cursor.getColumnIndex("mobph"))));
        telph.setText(modle.setName(cursor.getString(cursor.getColumnIndex("telph"))));
        addr.setText(modle.setName(cursor.getString(cursor.getColumnIndex("addr"))));
        email.setText(modle.setName(cursor.getString(cursor.getColumnIndex("email"))));
        firm.setText(modle.setName(cursor.getString(cursor.getColumnIndex("firm"))));
        blog.setText(modle.setName(cursor.getString(cursor.getColumnIndex("blog"))));
    }
}
```

点击编辑联系人按钮，跳转到编辑页面，附带联系人 id 和联系人头像 id 两个数据：

```
editbn.setOnClickListener(new OnClickListener(){
    public void onClick(View v){

        Intent intent=new Intent();
        intent.putExtra("id",passid );
        intent.putExtra("img", passing);
        intent.setClass(IContactShow.this, IContactEdit.class);
        startActivity(intent);
        finish();
    }
});
```

点击删除联系人按钮，弹出确认删除的对话框，用户可以选择删除或者放弃。删除操作是根据联系人的 id 从数据库中查找到并进行 delete 操作的。

```

final AlertDialog.Builder builder=new AlertDialog.Builder(this);
//删除键- 弹出对话框进行确认
delbn.setOnClickListener(new OnClickListener(){
    public void onClick(View v){
        builder.setTitle("warning")
        .setMessage("Are you sure to delete this CONTACT?")
        .setIcon(R.drawable.icon_warn)
        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {

@Override
    public void onClick(DialogInterface dialog, int which) {
        // TODO Auto-generated method stub
        //删除数据
        MyDBHelper helper= new MyDBHelper(IContactShow.this);
        SQLiteDatabase db=helper.getWritableDatabase();
        db.delete("icontactTbl", "_id=?", new String[]{passid});
        db.close();

        Toast toast = Toast.makeText(getApplicationContext(), "You have deleted a contact!", 200);
        toast.setGravity(Gravity.CENTER,0,0);

        LinearLayout toastView=(LinearLayout)toast.getView();
        ImageView iv=new ImageView(getApplicationContext());
        iv.setImageResource(R.drawable.icontact_1);
        //toastView.setBackgroundResource(R.drawable.icon_dlg);

        toastView.addView(iv);
        toast.show();

        Intent intent=new Intent();
        intent.setClass(IContactShow.this, IContactMainActivity.class);
        startActivity(intent);
        finish();
    }
});
delbn.setOnClickListener(new DialogInterface.OnClickListener(){
    public void onClick(DialogInterface dialog,int which){
    }
});
AlertDialog ad=builder.create();
ad.getWindow().setLayout(300, 200);
//ad.getWindow().setBackgroundDrawableResource(R.drawable.icon_dlg);
ad.show();
}
});

```

3.2.5 编辑联系人详细信息类 IContactEdit.java

编辑联系人页面由 IcontactShow.java 页面跳转而来，跳转过程中夹带了联系人 id 和联系人头像 id 的数据信息，数据的显示类似 show 界面，数据的修改和修改限制类似 add 界面。

编辑页面完成后可以选择保留编辑或者放弃更改。

点击保留修改按钮，数据库更新数据，跳转到主页面，显示新的数据信息。

```

okbn.setOnClickListener(new OnClickListener(){
    public void onClick(View v){

        //int imgid=imgv.getId();
        String sname=name.getText().toString();
        String snote=note.getText().toString();
        String smobph=mobph.getText().toString();
        String stelph=telph.getText().toString();
        String saddr=addr.getText().toString();
        String semail=email.getText().toString();
        String sfirm=firm.getText().toString();
        String sblog=blog.getText().toString();
    }
});

```

```

if(mobph.length()==11||telph.length()==11){
    ContentValues values=new ContentValues();
    values.put("img", imgID);
    values.put("name", sname);
    values.put("note", snote);
    values.put("mobph", smobph);
    values.put("telph", stelph);
    values.put("addr", saddr);
    values.put("email", semail);
    values.put("firm", sfirm);
    values.put("blog", sblog);
    MyDBHelper helper=new MyDBHelper(IContactEdit.this);
    SQLiteDatabase db=helper.getWritableDatabase();
    db.update("icontactTbl", values, "_id=?",new String[]{passid});
    db.close();
    Intent intent=new Intent();
    intent.setClass(IContactEdit.this, IContactMainActivity.class);
    startActivity(intent);
    finish();
}
else{
    Toast toast = Toast.makeText(getApplicationContext(), "Please enter 11 mobile phone number!", 30);
    toast.setGravity(Gravity.CENTER,0,0);
    LinearLayout toastView=(LinearLayout)toast.getView();
    ImageView iv=new ImageView(getApplicationContext());
    iv.setImageResource(R.drawable.icon_warn);
    toastView.addView(iv);

    toast.show();
}
}
});

```

若取消修改，则跳转回到之前的联系人详细信息显示界面，附带两个页面间传值。

```

clbn.setOnClickListener(new OnClickListener(){
    public void onClick(View v){

        Intent intent=new Intent();
        intent.putExtra("id", passid);
        intent.putExtra("img", imgID);
        intent.setClass(IContactEdit.this, IContactShow.class);
        startActivity(intent);
        finish();
    }
});

```

3.2.6 其他类简介

为了方便联系人数据的整体操作，添加了 Contact.java 类，用于传入和传出联系人信息值。

为了完成邮箱自动匹配功能，添加了 ImageAdapter.java 类，定义了头像数组，可以根据头像的数组 position 得到头像 id 数据传出。

```

public class ImageAdapter extends BaseAdapter
{
    // 定义Context
    private Context mContext;
    // 定义类型数组 即图片源
    private Integer[] mImageIds =
    {

        R.drawable.i1,
        R.drawable.i2,
        R.drawable.i3,
        R.drawable.i4,
        R.drawable.i5,
        R.drawable.i6,
        R.drawable.i7,
        R.drawable.i8,
        R.drawable.i9,

    };
}

```



```

// 获取图片ID
public long getItemId(int position)
{
    return mImageIds[position];//-----
}

public View getView(int position, View convertView, ViewGroup parent){
    ImageView imageview = new ImageView(mContext);
    // 给ImageView设置资源
    imageview.setImageResource(mImageIds[position]);
    // 设置布局 图片50x50显示
    imageview.setLayoutParams(new Gallery.LayoutParams(50, 50)); //宽度和高度
    // 设置显示比例类型
    imageview.setScaleType(ImageView.ScaleType.FIT_CENTER);
    return imageview;
}

```

四、实验遇到的问题及其解决方法

4.1 页面间传值问题

本次项目中有很多页面间传值的操作。

由于联系人 id 和联系人头像 id 均为整数，所以传递的值应当采取传整数的方法。但是由于在数据库查询语句 db.query() 函数的各项参数均为 String 类型数，而页面间传值后是根据联系人 id 进行搜索数据库找到对应联系人的，所以联系人 id 必须由整数转化为字符。又因为一方存整数一方取字符的话得到的结果是空，无法成功进行页面间传值，也就无法正确显示联系人信息。这个问题主要出现在 IContactMainActivity.java 到 IContactShow.java 和 IContactAdd.java 文件的跳转当中，show 到 edit 的跳转可以直接用传来的值继续传递下去。这里我采取的措施为，**存字符取字符**。IContactMainActivity 界面的 listview 分别显示了四项信息，联系人 id, 头像，姓名和电话，通过联系人 id 的 TextView 得到其中字符，再传递给下一个界面，这就解决了联系人 id 的传递问题。

```

TextView tvid = (TextView) view.findViewById(R.id.iid);
intent.putExtra("id", tvid.getText().toString());

```

这时候又有另一个问题，头像的 ImageView 对应的 id 和头像 id 不是一个，所以，传值时必须传递头像 id。但是在数据库查询到头像 id 后传递给了 ImageView 进行显示，无法直接从 ImageView 反得出头像资源的 id。这时，我采取的办法是**增加一个不显示的 TextView (iimgt)**，其值为头像 id，这样，可以和联系人 id 一样进行数据读取。由于头像 id 值传递过去后直接显示头像，不需要转化成字符，所以数据传递时是以整数的形式传递的。

```

TextView iv=(TextView)view.findViewById(R.id.iimgt);
intent.putExtra("img", Integer.parseInt(iv.getText().toString()));

```

在解决此问题的途中还遇到了另一个问题，主要是自己没搞清楚头像的 id 和头像在数组中的位置这两个不同概念。传值成功后，依据头像 id 直接赋值给一个 `ImageView` 即可正确显示头像，不应该拿来在头像数组当中找，这会造成访问不存在的空间而导致软件意外退出。

五、实验结论

在第一次试验基础上，本次实验加入了 `SQLite`，`Service`，和 `Intent` 操作，软件设计更加倾向于实用化，整体化。这次实验的 UI 为自主设计，仍然以简洁大方，轻松活泼为主旨，突出新颖个性的特色。

我首先针对试验要求进行规划，自己设计出了所有的 UI 界面和程序运转过程，在第一次的基础上，我对 `eclipse` 平台开发颇为了解，能够较为熟练地进行项目开发。首先编写好各个 `xml` 文件，然后依次编写各个类文件，逐步完成规划任务。

数据库操作是已经封装好的，所以调用起来较为容易，数据库的增、删、改、查功能都很方便地就能够实现，前提是查询语句书写正确，查询语句一定要注意空格的处理，多或者少都可能造成错误，无法正确进行数据查询。

这次实验在自主拓展功能上花费的时间较多，阅读了大量相关资料，耐心排查试验错误，最终取得了不错的实验成果。我认为 `eclipse` 的错误提醒机制非常有助于程序员检查错误，`Logcat` 窗口非常详细地记录了每次软件启动过程中的操作和错误记录，一旦发生软件意外中断，检查 `Logcat` 窗口的错误记录能够有效地定位错误点，有针对性地进行修改和调试，相当高效率。

这次的实验较上次更为细致，更加注重整体风格和用户体验。通讯录是非常普遍使用的手机软件，大家都有对于通讯录的使用经验，如何设计出简单易用而且吸引用户的通讯录是我这次的重要关注点。我根据自己的习惯，听取了一些人的建议意见，设计出了这款通讯录的 UI 和功能跳转，整体使用下来非常流畅，而且功能较为齐全。在按键和背景图案的选择上，我采取了主题制原则，选择相同风格的图案，尽量使软件看起来美观大方。

由于对于 `android` 开发的了解尚不够深刻，第一次做这样的实用软件，而且自己本身对 UI 设计也只是浅涉，`iContact` 肯定还有很多不足和需要改进的地方，希望您能提出宝贵的意见和建议，帮助我做的更好，谢谢！